Preprints of the 8th IFAC Symposium on Advanced Control of Chemical Processes
The International Federation of Automatic Control
Furama Riverfront, Singapore, July 10-13, 2012

A comparison of the computational efficiency of generalised function MPC using active set methods

B. Khan *, **J.A. Rossiter** *

* *Automatic Control and Systems Eng., University of Sheffield, UK*
e-mail: b.khan@sheffield.ac.uk., j.a.rossiter@sheffield.ac.uk

Abstract: This paper considers the computational efficiency of the recently developed generalised function parameterisations predictive control algorithms using active set methods. Alternative parameterisations have been shown to improve the volume of the feasible region when the number of degrees of freedom is limited. However, earlier work also demonstrates that some of the structure of optimisation problem is lost when using these parameterisations and thus requires more computation operations per step. This paper considers the dense problem structure arising from removing any redundant constraints and then considers the computational efficiency using the minimal constraint sets. Extensive simulation results suggest there can still be benefits from using more general parameterisations, although less significant then indicated by the number of degrees of freedom alone.

Keywords: Predictive control, Active set method, Computational efficiency, Feasibility, Generalised function parametrisations.

1. INTRODUCTION

Model Based Predictive Control (MPC) predictive control (Mayne et al., 2000; Rossiter, 2003; Camacho and Bordons, 2003) is the most common name for a computer control algorithms that uses past information of the inputs and outputs and a mathematical model of the plant to optimise its predicted future behaviour. MPC is well established and widely used both in industry and control research community and has reached a high degree of maturity. Much of current research is focused on extended the applicability of MPC to stochastic, nonlinear and robustness issues, but there is still substantial interest in fast optimisation or related computational aspects, even for the linear case. This is because all algorithms have to achieve a trade-off between feasibility, optimality and the expense of implementing the associated optimisation. Consequently, this paper explores the computational efficiency or benefits of deploying generalised function parameterisations within an optimal predictive control algorithm (Scokaert and Rawlings, 1998; Kouvaritakis et al., 1998).

The success of earlier industrial heuristic MPC algorithms motivated the research community to develop several algorithms with improved performance and feasible regions. There are several successful theoretical approaches but few of them are exploited commercially for real-time implementation. One important issue for real-time implementation is to solve an optimisation problem within the time determined by the sampling instant of the application and therefore the computational efficiency of an algorithm becomes critical. A trade-off has to be made between performance, feasibility and the computational burden when choosing from the currently available algorithms; in simple terms better feasibility usually implies a higher computational load or worse performance. It is also recognised that the range of industrial applications of MPC has been restricted in practice due to the computationally expensive on-line optimisation which is required. Recent work on explicit solutions to the constrained MPC problem formulation (Bemporad et al., 2002) significantly increase the potential application areas for MPC by reducing the online computational demand. However, the major drawback with these approaches is that the storage requirements, and online search, may grow exponentially with the optimisation variables, states and input dimensions, so that the 'explicit solution' is often only actually efficient for small problems (where dimension is no more than around 5) (Wang and Boyd, 2010). Consequently, this paper does not pursue the avenue of explicit solutions, but instead recognises the ongoing need to improve a more traditional MPC optimisation.

In predictive control, at each sample instant, a quadratic programming (QP) problem is solved to obtain a sequence of inputs and only the first input is applied to the plant. This process is repeated at every sample instant with a new state estimate. The scenario considered in this paper is one in which a QP problem has been solved by an active set method at the previous step and thus one only needs to solve a slightly perturbed QP from the preceding QP at the current time instant. In this case, active set methods are effective and the number of iterations required is typically a small polynomial dimension (Maes, 2010).

In this paper, the computational efficiency of generalised functions parameterisation within an optimal MPC is discussed, using both a generic optimiser and an active set method. The computational efficiency using online optimisation may be obtained by using a warm start strategy and

exploiting the structure of the QPs that rise in MPC problem formulation. More recently Laguerre, Kautz and generalised functions have been proposed as an effective mean of parameterising the input predictions using orthonormal functions (Khan and Rossiter, 2011b,c). Specifically it was shown that such parameterisation in general may improve the feasible region when the number of decision variables or degrees of freedom (d.o.f.) is limited. Nevertheless, one key question was still left unanswered: what is the computational efficiency using the online optimisation as the reduction in the number of degrees of freedom may be compromised by a loss of structure in the optimisation?

This paper develops on a recent contribution (Khan and Rossiter, 2011a) to this problem which explores the computational efficiency of the Laguerre optimal MPC. It was shown that some of the structure of the optimisation problem is lost when using Laguerre parameterisation. In contrast, Optimal MPC (OMPC) can has strong structure which can be exploited in the active set method thus allowing relatively inexpensive optimisation with large numbers of d.o.f.. Specifically, the aim of this paper is to extend this to generalised parameterisations and to consider a more holistic picture which allows the reduction of data storage requirements and the online computation time by removing any redundant constraints. The proposed procedure is based on standard OMPC, Laguerre OMPC, Kautz OMPC and generalised OMPC.

Section 2 will give the necessary background about problem formulation, predictive control, Laguerre optimal predictive control (LOMPC) and Kautz optimal predictive control (KOMPC). Section 3 presents generalised function parameterisation to optimal predictive control and proposed active set method based algorithm for GOMPC. Extensive simulation results with simulation setup are discussed in Section 4, showing the efficacy of the proposed algorithm and this is followed by conclusions and future work in Section 5.

2. BACKGROUND

This section will introduce the background information on MPC and assumptions used in this paper.

2.1 Problem formulation and Optimal MPC

Assume a discrete-time state-space model of the form:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \tag{1}$$

with $x_k \in \mathbb{R}^{n_x}$,$y_k \in \mathbb{R}^{n_y}$ and $u_k \in \mathbb{R}^{n_u}$ which are the state vectors, the measured output and the plant input respectively. Let the system be subject to constraints:

$$\begin{aligned} \underline{u} \leq u_k \leq \overline{u} \\ \underline{x} \leq x_k \leq \overline{x}. \end{aligned} \tag{2}$$

The performance index (Scokaert and Rawlings, 1998) to be minimized with respect to $u_k, u_{k+1}...$ is

$$\begin{aligned} J &= \sum_{i=0}^{\infty} (x_{k+i+1})^T Q (x_{k+i+1}) + (u_{k+i})^T R (u_{k+i}) \\ s.t. &\begin{cases} (1), (2) & \forall k \geq 0, \\ u_k = -Kx_k & \forall k \geq n_c \end{cases} \end{aligned} \tag{3}$$

with Q and R positive definite state and input cost weighting matrices. Where K is the optimal feedback gain minimizing J in the absence of constraints (2). In general a finite number, n_c, of free control moves are used (Rossiter, 2003) and thus satisfying (2) and (3) (Rossiter, 2003; Scokaert and Rawlings, 1998) implies that the state x_{n_c} must be contained in an invariant set χ_0 (maximum admissible set or MAS). The MAS is defined as

$$\begin{aligned} \chi_0 = \{x_0 \in \mathbb{R}^{n_x} \mid \underline{x} \leq x_k \leq \overline{x}, \\ \underline{u} \leq -Kx_k \leq \overline{u}, x_{k+1} = Ax_k + Bu_k, \forall k \geq 0\}. \end{aligned} \tag{4}$$

In compact form defined as $\chi_0 = \{x_k : Mx_k \leq b\}$ for suitable M and b.

For convenience, the degree of freedom can be reformulated in terms of a new variable c_k (Rossiter, 2003; Scokaert and Rawlings, 1998) which is the perturbation around the optimal unconstrained law required to satisfy constraints.

$$\begin{aligned} u_k &= -Kx_k + c_k, && k = 0, ..., n_c - 1, \\ u_k &= -Kx_k, && k \geq n_c, \end{aligned} \tag{5}$$

and hence the equivalent optimisation to (3) is

$$\min_C \; J_c = C^T S C \; s.t. \; Mx_k + NC \leq b; \tag{6}$$

where $C = [c_k^T, \ldots, c_{k+n_c}^T]$, for suitable S, matrices N, M and vector b (details available in the literature (Mayne et al., 2000; Rossiter, 2003; Gilbert and Tan, 1991)).

The maximal control admissible set (MCAS) χ_c, the feasible set for optimal control problem in (6) that satisfies all polytopic constraints, is defined as

$$\chi_c = \{x_k : \exists C, Mx_k + NC \leq b\}. \tag{7}$$

Algorithm 2.1. OMPC

$$c_k^* = arg \min_{c_k} \; J_c \quad s.t. \quad Mx_k + Nc_k \leq b;$$

Implement $u_k = -Kx_k + e_1^T c_k^*$, $e_1^T = [I, 0, \ldots, 0]$.

While OMPC handles constraints and allows the global optimal constrained performance, it may do so in only a small volume and in practice one might require a large n_c (d.o.f.) to obtain both good performance and a large feasible region. Consequently, alternative algorithms, discussed next, have been considered which reduce the value of n_c required to achieve a given feasible volume.

2.2 Laguerre OMPC (LOMPC)

This section will summarise the MPC algorithm (Rossiter and Wang, 2008) which uses Laguerre functions to represent the d.o.f.; these are defined network of discrete transfer functions as follows:

$$L_i(z) = \sqrt{(1-a^2)} \frac{(z^{-1} - a)^{i-1}}{(1 - az^{-1})^i}; \;\; 0 \leq a < 1. \tag{8}$$

The Laguerre functions can be computed using following state-space form (Rossiter and Wang, 2008)

$$L_{k+1} = A_L L_k, \tag{9}$$

where a is a parameter of choice to determine the desired time span of affect of the input perturbations c_k, and this lengthening as compared to the single perturbations deployed in OMPC, can enable the associated feasible region or MCAS to be larger (Rossiter et al., 2005; Rossiter

and Wang, 2008). The predictions for the LOMPC input perturbations are:

$$C = \begin{pmatrix} c_k \\ c_{k+1} \\ \vdots \end{pmatrix} = \begin{pmatrix} L(0)^T \\ L(1)^T \\ \vdots \end{pmatrix} \eta = H_L \eta \tag{10}$$

where η is the n_L dimension decision variable when one uses the first n_L column of H_L. The associated prediction cost and constraint set can be represented in term of η as

$$J_L = \eta^T [\sum_{i=0}^{\infty} A_L^i L(0) S L(0)^T (A_L^i)^T] \eta = \eta^T S_L \eta \tag{11}$$

$$M x_k + N H_L \eta \leq b \tag{12}$$

with $c_{k+i} = L_i^T \eta$, $L_i = A_L L_{i-1}$ and appropriate M, N.

Algorithm 2.2. LOMPC

$$\eta_k^* = arg \min_{\eta_k} \; J_L \quad s.t. \quad M x_k + N H_L \eta_k \leq b;$$

Define $c_k^* = H_L \eta_k^*$ and implement $u_k = -K x_k + e_1^T c_k^*$.

2.3 Kautz function parameterisation in OMPC

A Kautz function parameterisation as an obvious alternative to Laguerre within an optimal MPC (Khan and Rossiter, 2011b) and has more potential to improve feasibility and performance, for a given n_η. Kautz functions are defined using second order network as follows

$$\mathcal{K}_i(z) = \mathcal{K}_{i-1}(z) \frac{(z^{-1} - a)(z^{-1} - b)}{(1 - a z^{-1})(1 - b z^{-1})}; \tag{13}$$
$$0 \leq a < 1; \;\; 0 \leq b < 1.$$

With $\mathcal{K}_1(z) = \frac{\sqrt{(1-a^2)(1-b^2)}}{(1-az^{-1})(1-bz^{-1})}$. The main difference between Laguerre and Kautz function parameterisation is the definition of the perturbation signals i.e. $c_{k+i} = \mathcal{K}(i)^T \gamma$.

Algorithm 2.3. The KOMPC algorithm is summarized as

$$\gamma_k^* = arg \min_{\gamma_k} \; J_{KOMPC} \;\; s.t. \; M x_k + N H_K \gamma_k \leq b \tag{14}$$

Define $c_k^* = \mathcal{K}(0)^T \gamma_k^*$ and implement the control law

$$u_k = -K x_k + e_1^T c_k^*$$

The details of how to define J_{KOMPC}, H_K are omitted (Khan and Rossiter, 2011b).

2.4 Generalised function parameterisation in OMPC

Laguerre and Kautz are 1st and 2nd order parameterisations and thus it is logical to consider where higher order or generalised parameterisation techniques further improve the feasible region while maintaining performance. Generalised higher order orthonormal functions with 'n' pole (i.e. '$a_1, \ldots, a_n$') network define multiple time scales for the input perturbations to improve the feasibility without deploying large numbers of d.o.f. (n_c) (Khan and Rossiter, 2011c). Laguerre and Kautz functions are special cases of generalised functions. The generalised parameterisation (Khan and Rossiter, 2011c) is defined using a higher order network such as

$$G_i(z) = G_{i-1}(z) \frac{(z^{-1} - a_1) \ldots (z^{-1} - a_n)}{(1 - a_1 z^{-1}) \ldots (1 - a_n z^{-1})}; \tag{15}$$
$$0 \leq a_k < 1, \quad k = 1, \ldots, n$$

with $G_1(z) = \frac{\sqrt{(1-a_1^2)\ldots(1-a_n^2)}}{(1-a_1 z^{-1})\ldots(1-a_n z^{-1})}$. In discrete state space form $G_{k+1} = A_G G_k$.

The basic concept of using alternative parameterisation, not to model the input trajectories directly but rather the perturbations c_k around the unconstrained optimal is preserved. Hence the input prediction perturbation using a generalised function (Khan and Rossiter, 2011c) is given by

$$C = \begin{pmatrix} c_k \\ c_{k+1} \\ \vdots \end{pmatrix} = \begin{pmatrix} G(0)^T \\ G(1)^T \\ \vdots \end{pmatrix} \rho = H_G \rho \tag{16}$$

where ρ is the n_G dimension decision variable when using the first n_G column of H_G. The difference between Laguerre OMPC, Kautz and generalised OMPC is the H_G matrix. For further details readers are referred to (Khan and Rossiter, 2011b) and (Khan and Rossiter, 2011c). The prediction cost using generalised function parameterisation in terms of ρ is given by

$$J_G = \rho^T [\sum_{i=0}^{\infty} G(i) S G(i)^T] \rho = \rho^T S_\rho \rho \tag{17}$$

with $c_{k+i} = G(i)^T \rho$ (Khan and Rossiter, 2011c). The MCAS is calculated in a similar manner to Algorithm 2.2, the main differences in the calculation are the use of transformation matrix H_G instead of H_L.

Algorithm 2.4. The GOMPC algorithm is summarized as

$$\rho_k^* = arg \min_{\rho_k} \; J_G \;\; s.t. \; M x_k + N H_G \, \rho_k \leq b \tag{18}$$

Define $c_k^* = G(0)^T \rho_k^*$ and implement the control law

$$u_k = -K x_k + e_1^T c_k^*$$

Remark 1. It is straight forward to show, with conventional arguments, that all algorithms (i.e. LOMPC, KOMPC, GOMPC) give guaranteed recursive feasibility and stability in the nominal case whenever the set point is feasible. Moreover, it can be shown that the specific formulations are such that *the tail* of the optimum perturbation at sample k is a feasible, and often optimal, choice for the optimum pertubation at sample $k+1$.

3. GENERALISED FUNCTION PARAMETERISATION IN MPC USING ACTIVE SET METHODS

This section considers the optimisations for LOMPC, KOMPC and GOMPC in detail and in particular the potential, or not, to exploit any structure for an efficient active set method.

3.1 Optimisation Structure of GOMPC

The generalised function parameterisation approaches chooses generic 'stable' basic vector to match the trajectories of the global optimal MPC. These approaches reduce the d.o.f. to overcome the trade-off between feasible region and performance. The active set method (ASM) requires a computational time which is cubic in d.o.f. to solve the dense formulation in (6) and (18). To obtain both good performance and a large feasible volume, OMPC might require a large n_c (d.o.f.) which thus would compromise

the computational efficiency of an ASM. Various methods can be used to speed up the computation, for example one strategy is to exploit the structure of the QP that arises in a simple MPC formulation which includes all the constraints (Wang and Boyd, 2010), that is including the redundant ones. Of course this may requires more storage space with a large n_c for computations.

In generalised function parameterisation approaches the QPs become dense and the problem structure is less obvious, and thus a tailored ASM method is as yet not available with a consequent increase in computation operations per step for the same n_c as OMPC. In terms of online computation, OMPC requires more d.o.f. to improve the feasible region and the question is, should we use a parameterise MPC with a low n_c or add more d.o.f. to a conventional OMPC approach? In this paper focus will given on comparing the computational efficiency of the OMPC and parameterised algorithms. Critically however, as the formulation with GOMPC is already dense and there is no structural advantage in retaining redundant constraints and thus it makes sense to remove all the redundant constants before moving to the online computation load. This also reduces the storage requirement.

3.2 Active set method applied to GOMPC

An active set method (ASM) is introduced here to solve the optimisation problem in (18). An Active set method converts the proposed optimisation into a sequence of equalities problems (EPs), involving only equality constraints, which are solved to generate a sequence of iterates converging to the solution. At each iteration an active set method solves a Karush-Kuhn-Tucker (KKT) system defined by the 'proposed' active constraints. Consequently, when a significant number of constraints are active, the 'system' is much smaller than in an interior point method. Moreover, the active set approach is preferred here over interior point methods since it makes more efficient use of information from previous online optimisations, which provides a good initial guess, or warm start, for an active set.

The algorithm uses off-line computations to remove the redundant constraints to reduce the online computational load. This is especially desirable since the constraints matrices are by necessity dense (because the structure of the optimisation is lost when using function parameterisation). This become advantageous as the problem size decreases and overall online computational effort involved is comparable with that of lower dimensional problems. We demonstrates the resulting improvements in both computational burden and size of stabilizable sets (or feasible volume).

The following gives a brief description of an active set solver for the QP (18) and provides details of the computation involved in each step. Introducing the Lagrangian associated with the problem (18) can be defined as:

$$L(\rho,\lambda) = \frac{1}{2}\rho_k^T S_\rho \rho_k + \lambda^T(Mx_k + \widehat{N}\rho_k - b) \tag{19}$$

where $\widehat{N} = NH_G$. The KKT conditions of (18) are

$$S_\rho\rho_k + \widehat{N}^T\lambda = 0, \tag{20}$$
$$b - Mx_k - \widehat{N}\rho_k \geq 0, \tag{21}$$
$$(Mx_k + \widehat{N}\rho_k - b)^T\lambda = 0, \tag{22}$$
$$\lambda \geq 0 \tag{23}$$

The procedure is summarised as follows

Algorithm 3.1. Off-line tasks:
Removing redundant constraints (Kerrigan, 2000)

(1) Set $\tilde{As} = [M\ NH_G]$, $\tilde{bs} = b$ and $X = \begin{bmatrix} x \\ \rho \end{bmatrix}$. The problem is to remove all redundant inequalities in $\tilde{As}X \leq \tilde{bs}$ to obtain an irredundant description description $As \leq bs$ with same set results.
(2) Set $i = 1$, $As = [\,]$ and $bs = [\,]$.
(3) If $\max_X \tilde{As}_i X > \tilde{bs}_i$, then set $As = \begin{bmatrix} As \\ \tilde{As}_i \end{bmatrix}$ and $bs = \begin{bmatrix} bs \\ \tilde{bs}_i \end{bmatrix}$.
(4) If $i <$ length of $\tilde{As}$ then set $i = i+1$ and goto step 3, else terminate;
(5) The irredundant description is given by $AsX \leq bs$. The maximisation in step 3 can be implemented as a linear programming (LP).
$M = As_{(:,1:n_x)}$, $\hat{N} = As_{(:,n_x+1:end)}$ and $b = bs$.

On-line tasks:

(1) At first, check whether ρ_k is optimal in the subspace defined by W_k.
Then, define a move direction p and express J as a function of p:

$$\begin{aligned} f(\rho_k + p) &= \frac{1}{2}(\rho_k + p)^T S_\rho(\rho_k + p) \\ &= \frac{1}{2}p^T S_\rho p + p^T S_\rho \rho_k + const. \end{aligned}$$

Take a small step in direction p defined by the QP with equalities

$$\begin{aligned} \min_p\ & \frac{1}{2}p^T S_\rho p + p^T S_\rho \rho_k \\ s.t.\quad & \widehat{N}_k^T p = 0 \end{aligned} \tag{24}$$

The solution of the subproblem (18) is given by the solution of the KKT system

$$\begin{pmatrix} S_{\rho_k} & \widehat{N}_k^T \\ \widehat{N}_k & 0 \end{pmatrix}\begin{pmatrix} p \\ \zeta \end{pmatrix} = \begin{pmatrix} -S_{\rho_k}\rho_k \\ 0 \end{pmatrix} \tag{25}$$

(a) If the solution of (18) is $p = 0$ then ρ_k is optimal in the current subspace. Proceed to stage 2 below
(b) otherwise,

$$\rho_{k+1} = \rho_k + \alpha p,\ 0 < \alpha \leq 1. \tag{26}$$

The step size α must be chosen to maintain feasibility. So

$$\alpha = min\left(1,\ \min_{n_i^T p > 0} \frac{d_i - m_i^T x_j - n_i^T \rho_k}{n_i^T p}\right) \tag{27}$$

A constraint l which would yield $\alpha < 1$ in (27) is a blocking constraint. Add blocking constraint to the working set W_k to form W_{k+1} and update the iterate by (26). Otherwise, $\alpha = 1$ in (27), then working set, $W_{k+1} = W_k$.

Table 1. Complexity Vs Feasible Volume for $n_c = 2$

4-Dimensional Example				
Algo.	Ineq.	ASM (ms)	Quadprog (ms)	Feasible Vol.
OMPC	66	0.25	0.73	0.16
LOMPC	104	0.29	1.25	0.39
KOMPC	127	0.48	1.33	0.53
GOMPC	156	0.35	1.06	0.82
10-Dimensional Example				
Algo.	Ineq.	ASM (ms)	Quadprog (ms)	Feasible Vol.
OMPC	128	0.50	1.34	0.96
LOMPC	144	0.52	1.39	0.97
KOMPC	155	0.54	1.42	0.97
GOMPC	206	0.63	1.60	0.98
16-Dimensional Example				
Algo.	Ineq.	ASM (ms)	Quadprog (ms)	Feasible Vol.
OMPC	152	1.34	2.46	0.87
LOMPC	154	1.46	2.80	0.89
KOMPC	158	1.69	3.02	0.90
GOMPC	166	1.67	2.72	0.92
30-Dimensional Example				
Algo.	Ineq.	ASM (ms)	Quadprog (ms)	Feasible Vol.
OMPC	152	0.76	1.33	0.80
LOMPC	153	0.98	1.55	0.82
KOMPC	154	0.97	1.56	0.84
GOMPC	154	0.92	1.56	0.86

(2) An optimal ρ_k has been found for W_k, check optimality: set $\zeta_i^k = 0 \ \forall \ i \notin W_k$, the other Lagrange multipliers at this point are known from (25).
   (a) if $\zeta \geq 0$ then all KKT conditions hold and the optimal point has been found.
   (b) otherwise, there is a component $\zeta_q < 0$. Then W_{k+1} is formed by dropping q from W_k, and iteration repeated.

In an MPC formulation, the solution of one instance of a QP is close to the next instance, in the sense that some of the information gathered in the solution process may still be valid (Maes, 2010) [usually this will be *the tail*, that is the part not associated to the current sample]. A warm start strategy can be used to exploit an advanced starting point for solving the QP and thus to further improve the computational efficiency. This strategy may significantly reduce the number of iterations required and thus improve the computational effort.

4. NUMERICAL EXAMPLES

A simple implementation of the active set method is developed in MATLAB, which handles the case of a quadratic objective and box constraints. The main purpose is to compare the timing results for the alternative parameterisations algorithms and standard OMPC algorithm using ASM and quadprog.m methods on a 3.16 GHz Intel Core 2 Duo running Microsoft Windows XP. Our prime interest is to compare two aspects: (i) the size of the feasible regions; (ii) the complexity of the different algorithm solutions (in essence computation time) so that some comments can be made about the potential of the general function parameterisation approaches.

4.1 Simulation Setup

The optimal predictive controller with $n_c = 2$ is used as a basis for comparisons. All algorithms (OMPC, LOMPC, KOMPC and GOMPC) provide stability and feasibility properties but the volume of the maximum controller admissible control admissible set (MCAS) for each of them varies. Hence, it is necessary to compare both the complexity of the algorithms and the volumes of the associated MCAS. The inputs and states for all systems were constrained to $-0.08 \leq u \leq 0.08$ and $-2.4 \leq x_k \leq 2.4$, ($k = 4, 10, 16, 30$) with performance objective weighting matrices $R = I$ and $Q = C^T C$. For simplicity the Laguerre, Kautz and generalised functions were based on $a = 0.5$, $a = 0.6, b = 0.7$ and $[a_1, \ldots, a_4] = [0.75 \ 0.6 \ 0.7 \ 0.8]$.

4.2 Volume comparisons

The feasible volume is estimated using a large number of equi-spaced directions in the state space. For each direction, the distance from the origin to the boundary of the MCAS is determined; the larger the distance, better the feasibility. Finally these distances are normalised against the distance obtained with OMPC with $n_c = 15$, we realise this is somewhat arbitrary but it seems a pragmatic limit for the global feasible volume with sensible sampling and dynamics.

Feasible volumes are compared in table 1, 2 and figure 1 for random systems. Alternative parameterisations have noticeably better feasibility than OMPC and it is clear that GOMPC has a larger MCAS than OMPC, LOMPC and KOMPC. GOMPC gets within more than 82% of the global MCAS (OMPC with $n_c = 15$) with just a 4th order function dynamic parameterisation, and using $n_c = 2$.

4.3 Complexity comparisons

Table 1 list the computational time required to compute the control law as fixed number of d.o.f. (i.e. $n_c = 2$) for all randomly generated examples of different sizes using the active set method and generic optimiser solver (quadprog.m from MATLAB). It is clear that alternative parameterisation approaches requires more inequality constraints and increase the computational load around 10-20% using the active set method. Hence, generalised function parameterisation gives relative inexpensive optimisation to achieve good performance and a large feasible region.

The algorithm uses an off-line computation to reduce the problem size which becomes an advantageous to improve the computational efficiency of all the algorithms. This helps the standard OMPC to increase d.o.f. to improve both performance and feasible region.

4.4 Feasible volume vs computational load

Two random examples with 4 and 5 dimension system results are summarised in table 2. For 4 dimension system figure 1 shows the normalised feasible volume increases as varying the d.o.f. for all algorithms. GOMPC reaches global feasible volume with just 3 d.o.f. whereas, OMPC requires 15 d.o.f. Clearly alternative parameterisation algorithms required few number of d.o.f. to have global feasible region.

The computational load is compared with global feasible volume in table 2. An interesting observation is that the

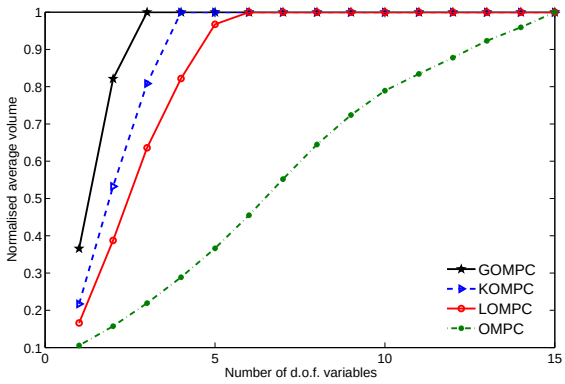

Fig. 1. Comparison of average relative radii of MCAS as total number of d.o.f. (n_c) vary for all four algorithms for 4-Dim. system.

Table 2. Complexity Vs Global Feasible Volume

4-Dimensional Example				
Algo.	Ineq.	ASM (ms)	Quadprog (ms)	n_c
OMPC	148	0.57	1.61	15
LOMPC	171	0.60	1.82	6
KOMPC	155	0.55	1.62	4
GOMPC	168	0.59	1.55	3
5-Dimensional Example				
Algo.	Ineq.	ASM (ms)	Quadprog (ms)	n_c
OMPC	201	0.68	1.69	15
LOMPC	198	0.68	1.72	7
KOMPC	199	0.55	1.62	5
GOMPC	202	0.67	1.66	3

primary factor to affect the computational load is due to the number of constraints rather d.o.f.. It is clear from table that alternative parameterisation approach requires same computational load with fewer d.o.f. as compare with OMPC. However generalised parameterisation may have tall-skinny matrix to improve the storage requirement.

Remark 2. These results are based on default choices for the parameters in GOMPC. Further improvements are possible by tailoring these parameters to the context.

5. CONCLUSION AND FUTURE WORK

This paper has carried out a preliminary investigation of the computational efficiency of generalised functions as an alternative parameterisation for improving the computational complexity in MPC algorithms with fixed number of d.o.f.. It is assumed that the structure for LOMPC, KOMPC and GOMPC is not easy to exploit and hence the first step, offline, is to remove all redundant inequalities and express the inequalities in a compact but dense format. In such a case, the remaining question is whether the dense form of OMPC with a high n_c can compete with the generalised function form with a low number of d.o.f.

Extensive simulation examples clearly re-affirm the message that for a fixed and low number of d.o.f., GOMPC, KOMPC and LOMPC give much better feasibility than OMPC. However, if instead one fixes the feasible volume to be similar, and use as many d.o.f. as are required, then one finds that OMPC may still be competitive in terms of computational load, despite requiring far more d.o.f. than for example GOMPC. Of course GOMPC will have a smaller data storage requirement associated to carrying fewer d.o.f. This suggests more work is required with a detailed focus on optimisation algorithms to determine to what extent the use of generalised functions are advantageous and specifically, to what extent the structure can be exploited to achieve some speed up.

For the future, another important area of interest concerns extensions to the uncertain case. For robust polyhedral sets there is a potential for rapid exponential growth in the number of implied inequalities. However, to the authors' knowledge no-one has yet looked at the potential role of generalised function parameterisations in this context and specifically whether some advantages arise.

REFERENCES

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Camacho, E. and Bordons, C. (2003). *Model predictive control*. Springer, London.

Gilbert, E. and Tan, K. (1991). Linear systems with state and control constraints: THE theory and application of maximal output admissible sets. *IEEE Trans. on Automatic Control*, 36(9), 1008–1020.

Kerrigan, E.C. (2000). *Robust Constraint Satisfaction:Invariant Sets and Predictive Control*. Ph.D. thesis, University of Cambridge.

Khan, B. and Rossiter, J.A. (2011a). Computational efficiency of Laguerre MPC using active set method. *Proc. IASTED ISC 2011*.

Khan, B. and Rossiter, J.A. (2011b). Exploiting Kautz functions to improve feasibility in MPC. *Proc. 18th IFAC World Congress*.

Khan, B. and Rossiter, J.A. (2011c). Generalised parameterisation for MPC. *Proc. IASTED ISC 2011*.

Kouvaritakis, B., Rossiter, J., and Cannon, M. (1998). Linear quadratic feasible predictive control. *Automatica*, 34(12), 1583–1592.

Maes, C.M. (2010). *A Regularized Active-Set Method for Sparse Convex Quadratic Prrogramming*. Ph.D. thesis, Standford University.

Mayne, D., Rawlings, J., Rao, C., and Scokaret, P. (2000). Constrained model predictive control. *Automatica*, 36(6), 789–814.

Rossiter, J. (2003). *Model-based predictive control, a practical approach*. CRC Press, London.

Rossiter, J., Kouvaritakis, B., and Cannon, M. (2005). An algorithm for reducing complexity in parametric predictive control. *IJC*, 78(18), 1511–1520.

Rossiter, J. and Wang, L. (2008). Exploiting Laguerre functions to improve the feasibility/performance compromise in MPC. *Proc. CDC*.

Scokaert, P. and Rawlings, J. (1998). Constrained linear quadratic regulation. *IEEE Trans. on Automatic Control*, 43(8), 1163–1168.

Wang, Y. and Boyd, S. (2010). Fast model predictive control using online optimization. *IEEE Trans. on Control Systems Technology*, 18, 267–278.